TDB-ACC-NO: NB8909143

DISCLOSURE TITLE: Direct Memory Access TAG Architecture

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, September 1989, US

VOLUME NUMBER: 32

ISSUE NUMBER: 4B

PAGE NUMBER: 143 - 151

PUBLICATION-DATE: September 1, 1989 (19890901)

CROSS REFERENCE: 0018-8689-32-4B-143

DISCLOSURE TEXT:

- A method is described for increasing available Direct Memory Access (DMA) channel bandwidth on a switched channel.
- A method is described that permits a DMA channel to be serially reusable (without software intervention) in an environment where multiple DMA requests are outstanding while providing a high degree of system protection. During a period of no DMA data traffic, the channel is not bound to a particular device (as in conventional DMA channel architecture) but is free to accept data from any authorized device. ***** SEE ORIGINAL DOCUMENT *****

The method uses the notion of a `tag' (or ticket) in conjunction with the address of the source device to grant and validate transmission authority to this system node. The TAG gives permission to a specific node to transmit data to this node. The TAG forms an association between: a 'block' of data, the storage address for that block, and the specific source node. ***** SEE ORIGINAL DOCUMENT.

- A single request of data may require the transmission of multiple `blocks' of data from a given source and thus multiple TAGS (permission tickets).
- Since the source address is used in the association process, a form of protection is built into the architecture. That is, one source can not misrepresent itself as a different source. The validity of the source is guaranteed by the interconnecting switch. The TAG is a ticket for the passage of one data block from one source to one destination. The destination assigns TAGS to be used to transmit data to it and therefore paces the arrival of data. A TAG is placed in a TAG table when it is issued and is invalidated (punched) when it is used. This means that the TAG (ticket) can only be used once. The

TAG table also contains the storage address to be used to receive data. The TAG is the ticket for one data block to traverse from one source to one destination.

It does not contain the routing.

This simple architecture defines address bounds, pacing, and addresses system integrity.

- The method may be used where the system consists of multiple processors, workstations, and control units interconnected through a central switch via a high speed serial link, as shown in Fig. 1.
- Data and control information is transferred between the system nodes through BLK and MSG hardware adapters, respectively:

.BLK - used to transfer large blocks of data.
MSG - used to transfer messages or control information consisting of 1-256 bytes.

- Both the BLK and MSG facilities utilize a serial link and a switch port, as shown in Fig. 2. The method provides the system with the following advantages:

The ability to associate a given data block with a specific address.

- The ability to receive 'out of order' data. That is, to receive data for an entire request or data 'blocks' (pages) of a request in any order.
- A transmitting node is not required to have knowledge of the address map of the receiver.
- Furthermore, the TAG architecture prevents an address from being supplied by any external source.
- Since the TAG represents permission to a specific system node to transmit data, it can also filter out unauthorized transmissions.
- The BLK channel is bound to the data transfer only at the point of actual data transfer and not at the time of the request.
- The DMA TAG architecture is based on the following PRINCIPLES:

STORAGE ADDRESS HELD WITHIN NODE

A storage address has meaning only within that node. The ability to supply a specific storage address from a facility external to that node is not permitted.

PERMISSION TO TRANSFER

Permission is required from the 'receiver' before a data block can be transmitted. Permission is granted through the use of TAGS.

- The DMA TAG architecture utilizes the following STRUCTURES:

TAG

A TAG is the ticket or permission by a receiver to a specific transmitter to transmit a data block.

- Each TAG is associated with both a given piece of data and a specific main storage address. Each TAG represents permission to transmit one physical transport layer 'block' of data. Multiple TAGS would be allocated and given to a specific node in connection with a data request (read or write).

TAG TABLE

The TAG table is used by the receiver to associate a TAG received with on a BLK transfer with a main storage address for the associated data. The TAG table is not only used to determine the address at which the data is to be stored at, but also to verify the authority of this transmitting node to send data.

RING QUEUE

A ring-queue mechanism is used by both the transmitter and receiver.

From the prospective of the transmitter, software places data blocks on the queue when they are ready to transmit and the BLK hardware removes them off the queue when a block has been successfully transferred. The transmit BLK hardware also posts block completion status on a Status queue in response to a BLK ACK by the receiver. The receive BLK hardware will post the successful receipt of each BLK on a ring-Q.

- Ring-Queues used by the BLK facility:
  Transmit Data
  Transmit Status
  Receive Status

PACING: Pacing refers to the ability of a given node involved in a data transfer to control the rate or pace at which data blocks are flowing over the serial link. The DMA TAG architecture provides the ability for both the receiver and transmitter to pace the operation.

- Pacing by the receiver is controlled through the TAG mechanism. Permission to transfer a given number of 'pages' or 'blocks' is granted to the transmitter in the form of TAGS. The number of TAGs given to the transmitter indicated the number of 1KB blocks it has permission to transfer. It also represents the number of blocks of main storage that has been allocated by the receiver. This represents a guarantee by the receiver that it will be capable of receiving that amount of data.
- Pacing is controlled by the transmitter on the basis of its ability to assemble the requested data. From the view of the serial link, pacing is controller by the transmitter through its ability to post blocks on the Transmit Data ring-Q.
- READ FLOW: The read operation is initiated by a request to read data (multiple pages) from some I/O facility. The BLK device driver will then:

    allocate the required number of main storage
    pages.
- build the TAG Table

    Associate the main storage real address with a
    TAG (4 TAGS per page)
    build the 'requesting' MSG
    the request (record, track, cylinders, etc.)
    TAGS (IN ORDER RELATIVE TO THE REQUEST OF
    THE ASSOCIATED
    data block).

- The receipt of the MSG will initiate the required action by the I/O facility. It will access the device, associate the appropriate TAGS with the data and transmit the data to the requester. This process will continue until the entire request has been satisfied and will end with the transmission of a status message.
- Depending on the characteristics of the device and design of the system node, data blocks may be transmitted in an order other than requested. The system node may choose to accumulate some number of data blocks before data is transmitted over the serial link to improve the effective band width of the link.
- The DMA TAG architecture accommodates this 'out of order' condition through the use of the TAGS as follows:

TRANSMITTER

The receiver built the initial 'request' MSG by maintaining the same relative position of each TAG as its associated data within the request list.

- As the transmitter accumulates the data blocks, it will re-associate that block with its TAG. The transfer will be initiated by simply placing the address and control information (with the TAG) of the block(s) on the Transmit Data Ring-Q.

RECEIVER

The header of the incoming block will contain the TAG of that block. The receiver will validate the block by determining if this TAG is valid for this source (transmitter) by accessing the TAG Table.

- If this is a valid BLK transfer, the TAG table will supply the real main storage address for this block. If this is an invalid BLK, the receiver will simply reject the block.
- It will access the device (i.e., file) and retrieve data in the order appropriate to the device (i.e., data or records may be obtained from the device in an order other then requested).
- A Floating Channel architecture, shown in Fig. 3, provides the system with a hardware mechanism to automatically select an alternate

path if the primary path is unavailable.

- The Floating Channel architecture places an additional requirement on the TAG table in that it must be common to all BLK facilities within the local node.

- The Floating Channel architecture assumes that each system node (processor or control unit) may have multiple BLK facilities to increase both the aggregate bandwidth and lower latency. Each of the BLK facilities are operating under control of a single (common) TAG table as defined in the DMA TAG architecture. The Floating Channel architecture requires that this TAG table be addressable by ALL BLK facilities within a given node.

- A request is made via the MSG facility. The request will contain:

  a description of the requested data (on a read, for example),
  a TAG for each block of data, and
  a list of the alternate paths (or a table containing alternate paths for all system nodes).

- When a data source node has accumulated one or more blocks of data, it will attempt to transmit the data over one (primary) path. If that path is busy, an alternate path will be tried until a path can be established. The Floating Channel architecture provides the ability to receive the inbound data on ANY of the BLK facilities. The TAG provides permission to receive the data and the association of the data with the storage address.

- The level of alternate pathing being defined is shown in Fig. 3. A request to transmit a block(s) of data is assigned (bound) to a given BLK unit by software, for example, through a ring-Q update process. The task of selecting an alternate path (if/when required) is confined to the scope of the switch unit this BLK unit is attached to. As shown in Fig. 3, for data flowing from Node 1 to Node 2 alternate paths 'A/J' and 'A/K' are available.

- The Floating Channel architecture is based on the following data structures:

ALIAS NAME TABLE: The Alias Name table provides the means to associate the 'logical' address or ID of a given node with a set of 'physical' alias IDs for that node. This will permit, for example, a node to give TAGS (transmit authority) to a 'logical' al' node ID and accept transmit data from any of its alias BLK IDs. In Fig. 3 the following Alias Name table entries would exist:

Node 1
'A' 'B' through Switch 'X'

'C' 'D' through Switch 'Y'
Node 2
'J' 'K' through Switch 'X'
'L' through Switch 'Y'

TAG TABLE: A TAG table common to all BLK units within a given system node is used in the Floating Channel architecture.
This
delays the binding of a given data transfer to a physical BLK unit until receipt of the 'Start-of-Frame' character of the transmission. This preserves the DMA TAG architecture principles of:

storage addresses held within a node,
permission to transfer, and
floating channels.

- The Floating Channel architecture assumes the following system configurations:

BLK CHANNELS

Multiple BLK channels may exist in each node of the system, providing the opportunity for alternate paths to that node.

TAG TABLE

The TAG table must be housed in a location that is accessible by all BLK units that are considered as having alias IDs. This will permit them to be able to validate the authority of the incoming data and load it into storage.

ALIAS NAME TABLE

The Alias Name table must be housed in a location that is accessible by all BLK units that are considered as having alias IDs. This will permit them to be able to identify the 'logical' class of a given alias ID.

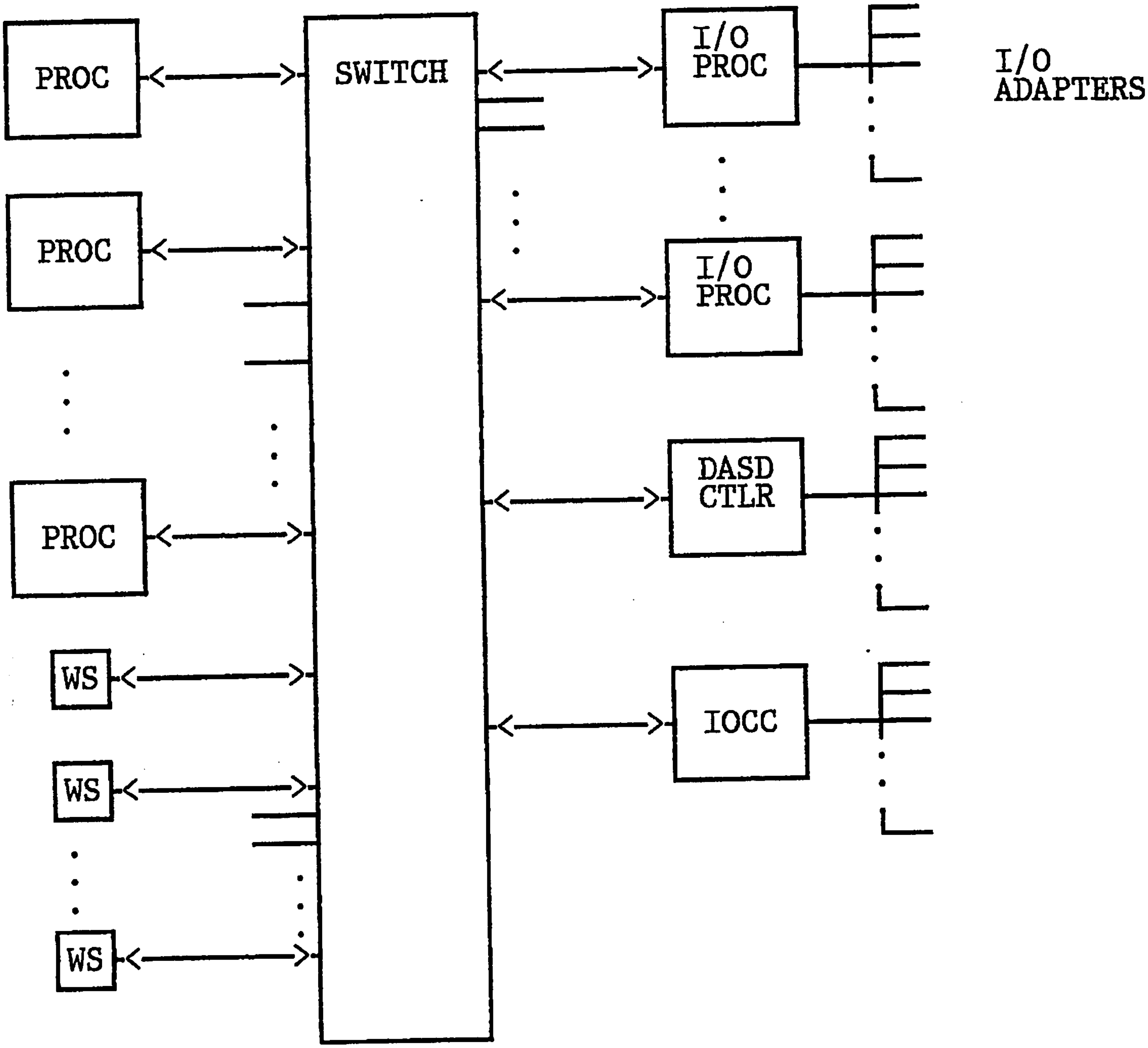
PROC
PROC
PROC
WS
WS
WS
SWITCH
I/O PROC
I/O PROC
DASD CTLR
IOCC
I/O ADAPTERS

FIG. 1

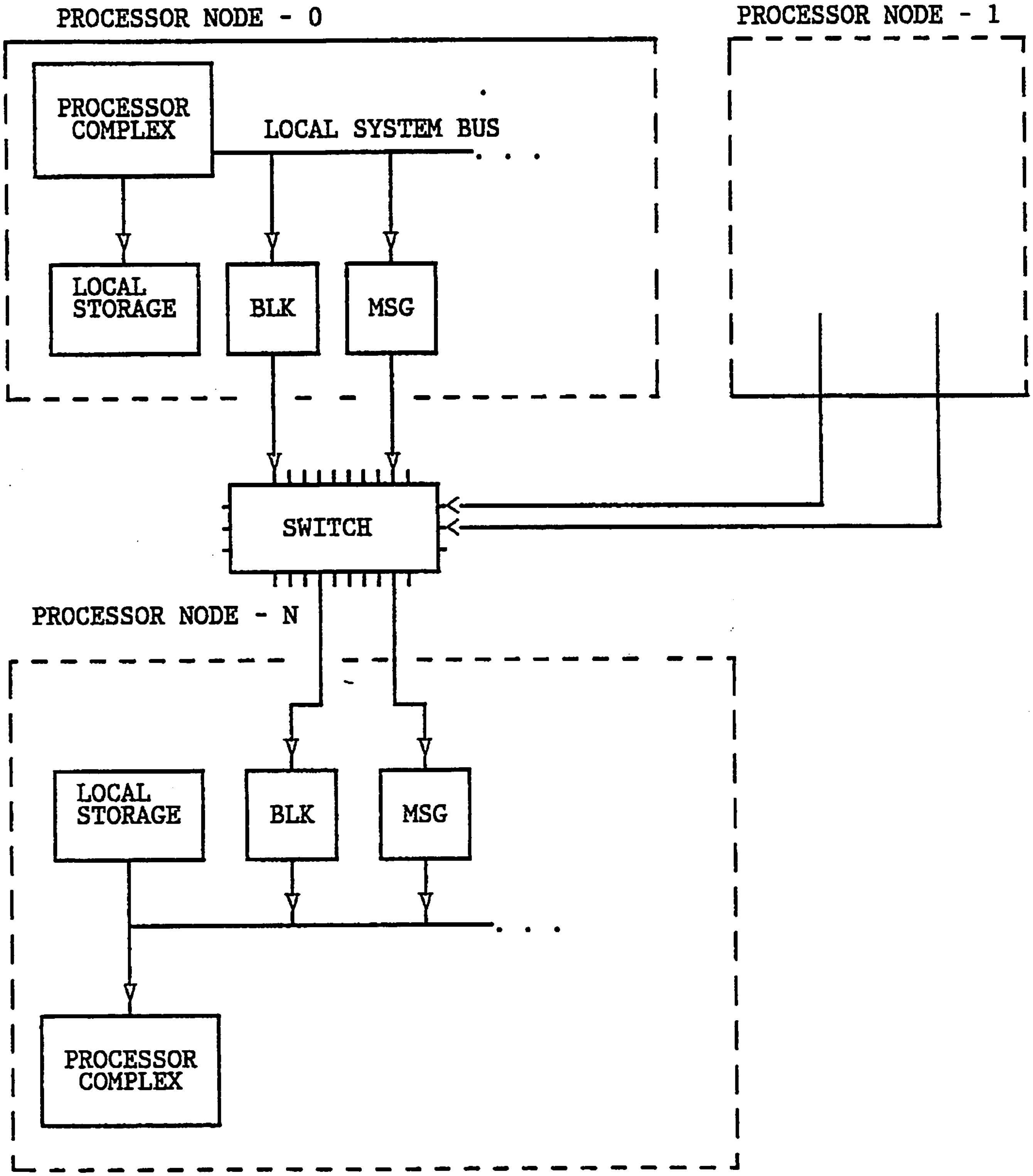
PROCESSOR NODE - 0
PROCESSOR NODE - 1
PROCESSOR COMPLEX
LOCAL SYSTEM BUS
LOCAL STORAGE
BLK
MSG
SWITCH
PROCESSOR NODE - N
LOCAL STORAGE
BLK
MSG
PROCESSOR COMPLEX

FIG. 2

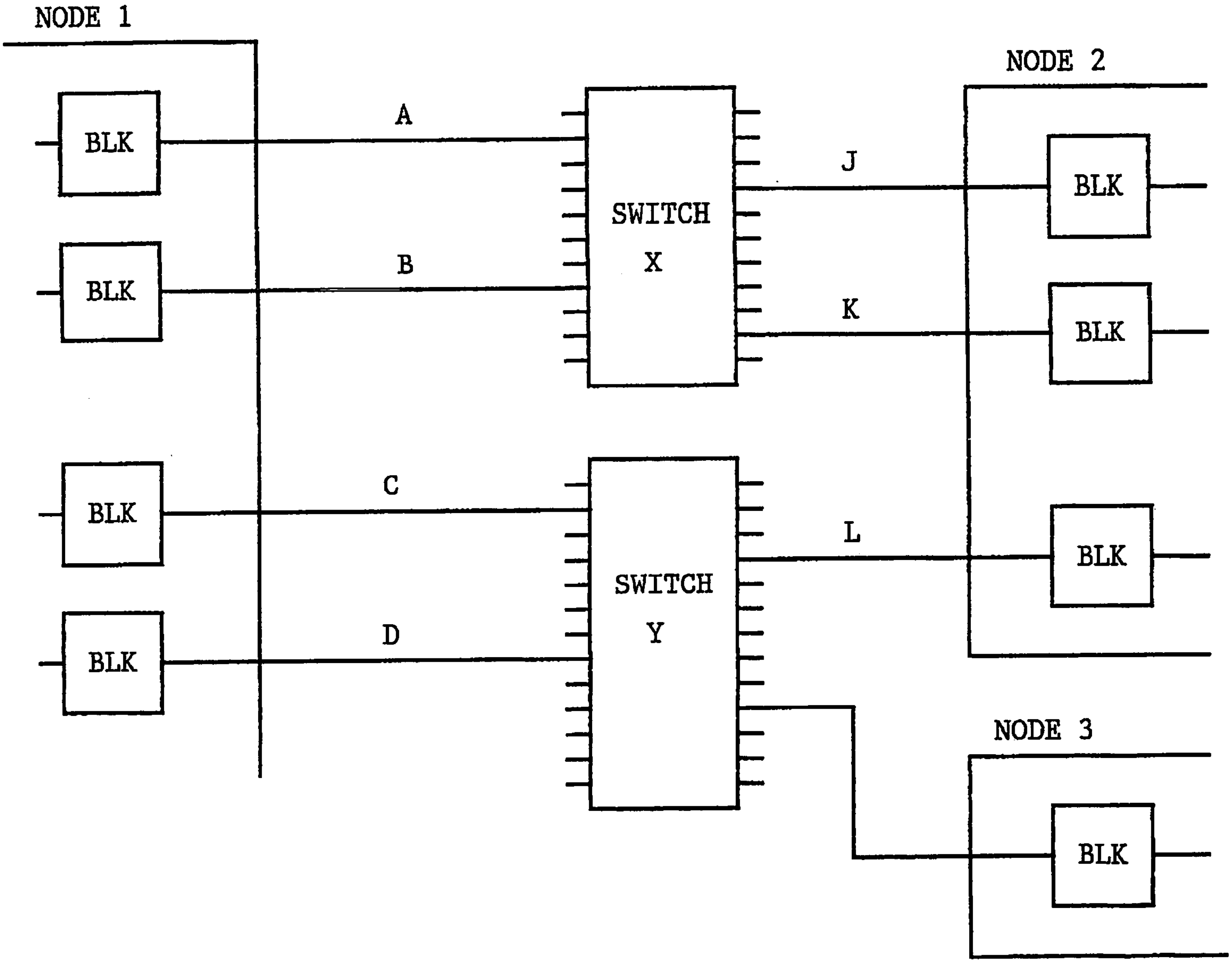
NODE 1
BLK
BLK
BLK
BLK
A
B
C
D
SWITCH
X
SWITCH
Y
J
K
L
NODE 2
BLK
BLK
BLK
NODE 3
BLK

FIG. 3